

NETWORK ANALYSIS FOR GPS UTMNAV SYSTEM: PRELIMINARY STAGE

Siti Nuruljannah A. Majid, Tajul A. Musa, Halim Setan, Ivin A. Musliman

Department of Geomatic Engineering
Faculty of Geoinformation Engineering and Sciences
Universiti Teknologi Malaysia (UTM) Skudai Malaysia
e-mail:
snuruljannah2@siswa.utm.my
Tel: +6075530801
Fax: +6075566163

Abstract

This ongoing research concentrates on the network analysis of the existing UTM navigation system (UTMnav). The aim is to overcome the limitation of UTMnav system in network analysis. A few shortest-path algorithms have been studied namely Dijkstra's, Floyd Warshall and A-Star (A). Each algorithm was reviewed and a simulation test was conducted to evaluate the selected algorithms. The simulation test was conducted using Visual Basic (VB) environment. A database of the selected network is created using MapInfo Professional and AutoCAD Map software. The understanding of the simulation test will motivate this research to improve on the UTMnav network analysis.*

1. INTRODUCTION

Nowadays, navigation system has become an information system in the sense of decision support. This is the reason why, it receives major attention from the mass market. Navigation system consists of two fundamental components: positioning and guidance. Therefore, the previous UTMnav developed by Ng (2008) in VB environment have three modules in order to provide positioning and guidance facilities. The three modules of the UTMnav are 1) database; 2) GPS-GIS integration and, 3) road network analysis.

The network analysis is the core of navigation system since it provides guidance facilities. It is based on a vector model of the navigational infrastructures. Methods of modeling the real world in general and the navigational infrastructure in particular apply well-known concepts as being used in geographical information system (GIS). The main modeling components are geometry and thematic information (semantics). Geometry can be divided into two which are matrices and topology. For instance, being connected or being adjacent, or intersecting each other and containing other objects. Despite the fact that semantics describe all thematic information, in the case of navigation systems it makes sense to distinguish between node– specific semantics (e.g., traffic light), node-to-node (edge) specific semantics (e.g., road class), and edge-to-edge (relation) specific semantics (e.g., turn restrictions) (Hofmann *et al.*, 2003). Modeling the navigational networks (e.g., road networks) is strongly correlated with graph theory, which mathematically deals with nodes-edges structures. The shortest-path algorithm is a type of graph search-algorithm.

There are numerous of graph search algorithms that can be implemented on the graph to identify the optimal edge in the network such as Dijkstra, Floyd Warshall, A* and Bellman. The efficiency of those algorithms is considerably different by its average execution time and the type of navigational networks, in term of network size and topology characteristics. Therefore, it is necessary to carry out the simulation test concern on which shortest-path algorithm is most appropriate to improve the network analysis of the UTMnav. The proposed algorithms of this study was Dijkstra, Floyd Warshall and A*.

The remainder of this paper is structured as follows; section 2 reviews the related works on the database and shortest-path algorithms. Next, section 3 introduces brief concept of the selected algorithms. In section 4, the design and development of the database are explained. A part of the simulation of the algorithm in VB environment is presented in the section 5. Finally, the conclusion of the paper is presented in section 6.

2. RELATED WORKS

So *et al.* (2005) proposed the spatial indexing method to reduce the map description data and increase user understandings of the map. In order to reduce the size of map data, firstly, the data is aligned by importance and incrementally presented to the user by its importance. In addition, the size of the map data can be reduced by using symbols which represent data in a concise manner. Second, to increase the user understandings about route (optimal path showing start and destination) and its surroundings, a new method which converts a complex route and its surroundings into a simplified sketch map. This process is also suggested to correct distortion which occurs during the sketch map generation process. The indexing method supports all the map generalization operations and manipulates zoom in-out quickly by leveling the generalized data. Studies of shortest-path algorithms have been concerning about the efficiency of those algorithms to the network.

Shortest-path algorithms are a type of graph search algorithms. It's well known as a fundamental component of network models. Hence, although the implementation of graph search algorithms is complex but it's still widely studied and new contributions keep appearing in the wide range application. Nadi and Delavar (2008) proposed to employ A* shortest-path algorithm. They claimed that this algorithm considers fewer nodes than any other searching algorithm with the same heuristic. The A* works based on an 'optimistic' estimate. The cost of a path from the origin to the destination will be almost the same to the estimation. Therefore, they used to implement this shortest-path algorithm with some assumption to reduce the complexity of routing algorithms.

In the other hand, Xu and Chen (2004) claimed that the Dijkstra's shortest-path algorithm is for computing the shortest-paths from single source node to every other node in a weighted graph and this algorithm computes the shortest-path tree containing approximately 50% of nodes in a graph before the shortest-path between a source node and a destination node was found. Anjang (2001) and Ng (2008) conducted their project on the analysis of the shortest-path algorithm for vehicle navigation system. They choose to develop the shortest-path algorithm based on Floyd Warshall algorithm. Floyd Warshall shortest-path algorithm was designed based on the all-pairs shortest-path problem.

3. THE SHORTEST-PATH ALGORITHMS

In designing the networks for navigational purposes, many shortest-path problems occur for instance to find the shortest-path between two given nodes of a graph, finding the shortest-paths from a given node to all other nodes, and finding the shortest-paths between all pairs of nodes. Thus, the shortest-path algorithms are designed to determine the shortest distance from the source node to every other node in the graph (Alsuwail, 1999).

3.1 The Dijkstra's Algorithm

Among the numerous shortest-path algorithms, Dijkstra's shortest-path algorithm is currently the most commonly used in GIS systems today. Not only it is a popular shortest-path algorithm in GIS systems, yet widely used in programs made for network modeling. The biggest drawback to Dijkstra's algorithm is that it does not work with negative lengths (Hofmann *et al.*, 2003). It turns out that one can find the shortest-paths from a given source to all nodes in a graph at the same time, hence this problem is sometimes called the single-source shortest-path problem. It tends to be relatively fast and efficient compared to other algorithms in solving the same problem.

The basic idea of Dijkstra's algorithm is that it generates a tree, which grows iteratively by creating one labeled node after the other and by branching out radially with the start node as a root. As it is known from the breadth-first search algorithms, in an idealized case, the tree is extending symmetrically if all the arcs have equal costs (Earnshaw, 1985). A standard implementation of Dijkstra's algorithm leads to a complexity of $O(n^2)$, where n is the number of nodes in the network. Dijkstra's algorithm implements the label setting procedure. Label setting approach assigns a permanent label once it has been set permanent. It makes Dijkstra's algorithm more efficient with the respect to the time complexity.

3.2 The Floyd Warshall Algorithm

Floyd's algorithm solves the all pairs shortest-path problem in the diagraph. In the algorithm's process, it looks for the best yet path to find a final path that is better than all others. Depending on the size of the graph this algorithm may take more time than Dijkstra's algorithm but has the potential to find a solution as good as or better than Dijkstra's algorithm (Cormen *et al.*, 1989). The main constraint of applying this algorithm is it burdens the user with $O(n^2)$ for space and $O(n^3)$ for time complexity where n is the number of nodes in the network.

Floyd's algorithm used the contrast searching strategies compared to the Dijkstra's. Floyd's algorithm applies the depth-first search. The tree may extend predominantly into a certain destination and much closer to a shortest-path result (Knuth, 1973). Besides, Floyd's algorithm is based on the label-setting algorithm that makes Floyd's algorithm still working even if on the negatives arc lengths.

3.3 The A-Star Algorithm

The A* algorithm is also a commonly used algorithm for shortest-path problems. When implemented, the A* algorithm can find a solution as good as Dijkstra's with less of a burden on memory (Russell & Norvig, 2003). The main reason why it is not used as much as Dijkstra's algorithm is because it tends to be much more difficult to implement when used to solve a shortest-path problem.

The A* algorithm is a frequently used implementation of a heuristic shortest-path approach. Heuristic methods are a powerful tool to improve the efficiency of the algorithm and are very well suited for combination with other practical shortest-path methods (Pearl, 1984). A standard implementation of A* algorithm $O(n)$ for space and $O((E + n) \log n)$ for time complexity which n indicate the number of nodes in the network.

4. UTMnav: DATABASE DESIGN AND DEVELOPMENT

The basis of spatial database in UTMnav is obtained from digital map in RSO coordinate system (Ali, 2007). **Figure 1** shows part of the digital map of UTM campus area. The spatial editing was conducted using the AutoCAD Map and the digital map was exported into MapInfo software. The attribute data of the digital map was assigned using the MapInfo software.

The Universal Translator (see **Figure 2**) is used to provide attribute data table from the digital map which is stored in *.tab file format. The number and name of fields for the attribute data entry depend on the requirement of the system design. For instance, the designation of the road centerline, it is necessary for the database to have the field "name_of_road", "length_of_road, from", "to", "_from" and "_to". **Figure 3** shows the example of designation for road centerline database. Meanwhile **Figure 4** shows the example of the attribute data that assign for spatial data.



Figure 1: Part of the digital map of UTM campus area

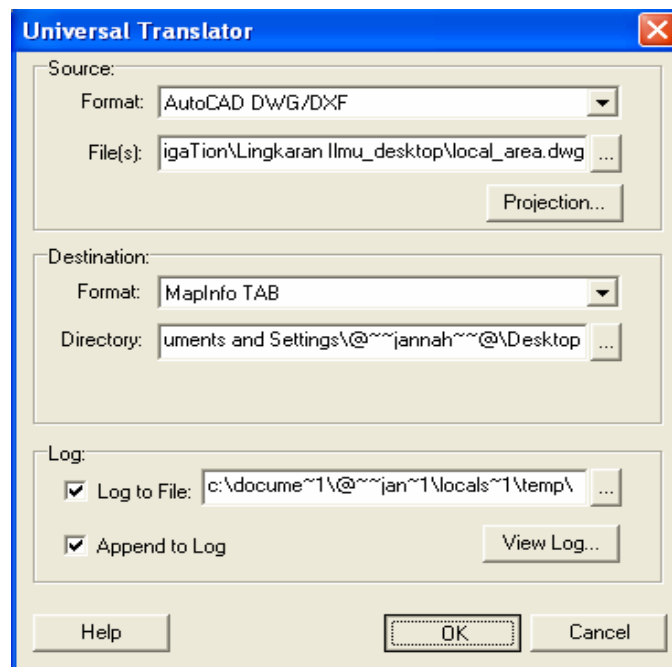


Figure 2: The window of Universal Translator

Fields	Type	Indexed
ID	Integer	<input type="checkbox"/>
name_of_road	Character(30)	<input type="checkbox"/>
lenght_of_road	Decimal(8,3)	<input type="checkbox"/>
from	Integer	<input type="checkbox"/>
to	Integer	<input type="checkbox"/>
_from	Integer	<input type="checkbox"/>
_to	Integer	<input type="checkbox"/>

Field Information

Name:

Type:

☒ Table is Mappable

Projection...

OK Cancel Help

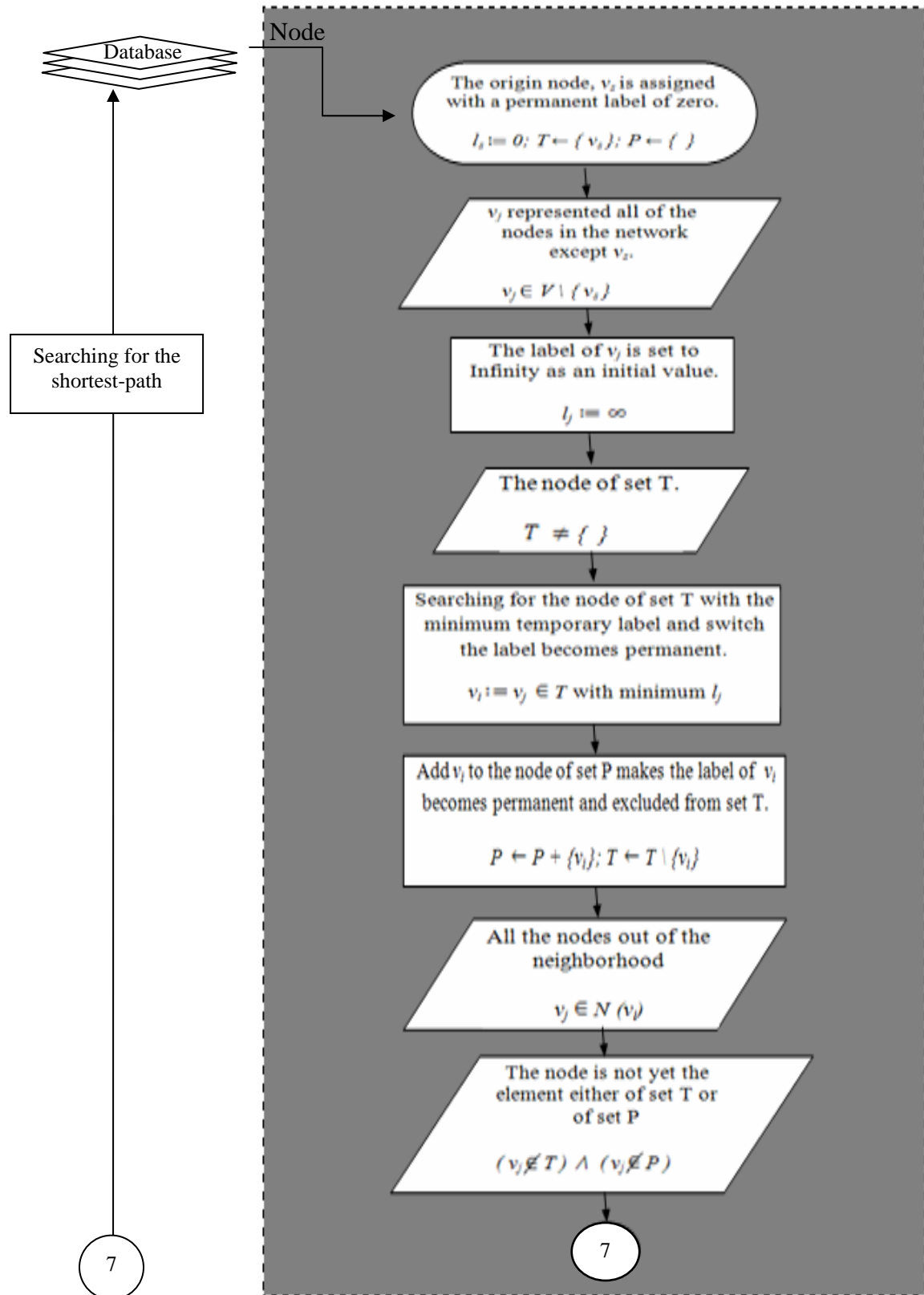
Figure 3: Fields that require in providing database for centerline

ID	name_of_road	lenght_of_road	from	to	_from	_to
3	jalan_tawakal1	0.000	3	4	4	3
4	jalan_tawakal2	0.000	4	5	5	4
5	jalan_amal1	0.000	5	6	6	5
6	jalan_amal2	0.000	6	7	7	6
7	jalan_hikmah2	0.000	7	8	0	0
8	jalan_taqwa1	0.000	8	9	9	8
9	lingkaran_ilmu1	0.000	9	10	10	9
10	lingkaran_ilmu2	0.000	10	11	11	10
11	lingkaran_ilmu3	0.000	11	12	0	0
12	lingkaran_ilmu4	0.000	12	13	13	12
13	jalan_budi1	0.000	13	14	14	13
14	jalan_budi2	0.000	14	15	15	14
15	jalan_budi3	0.000	15	16	0	0

Figure 4: Example of the attribute data assign for spatial data respectively

5. THE 'DIJKSTRA' SIMULATION TEST & ANALYSIS

The simulation test is carried out to evaluate the Dijkstra's algorithm. A simulation code which is written in Visual Basic (VB) programming language has been used. The flowchart in **Figure 5** shows the methodology of implementing the Dijkstra's algorithm.



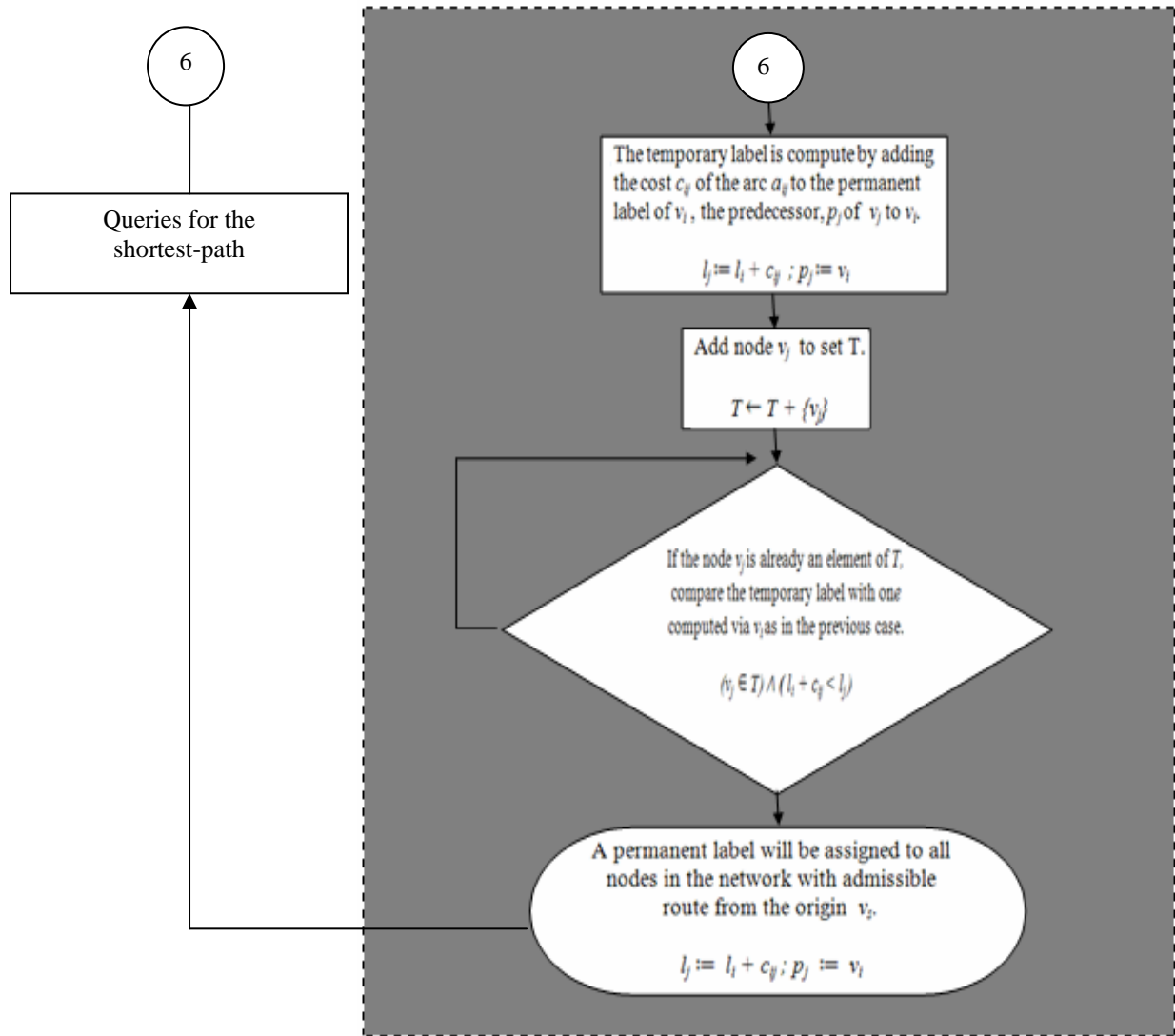


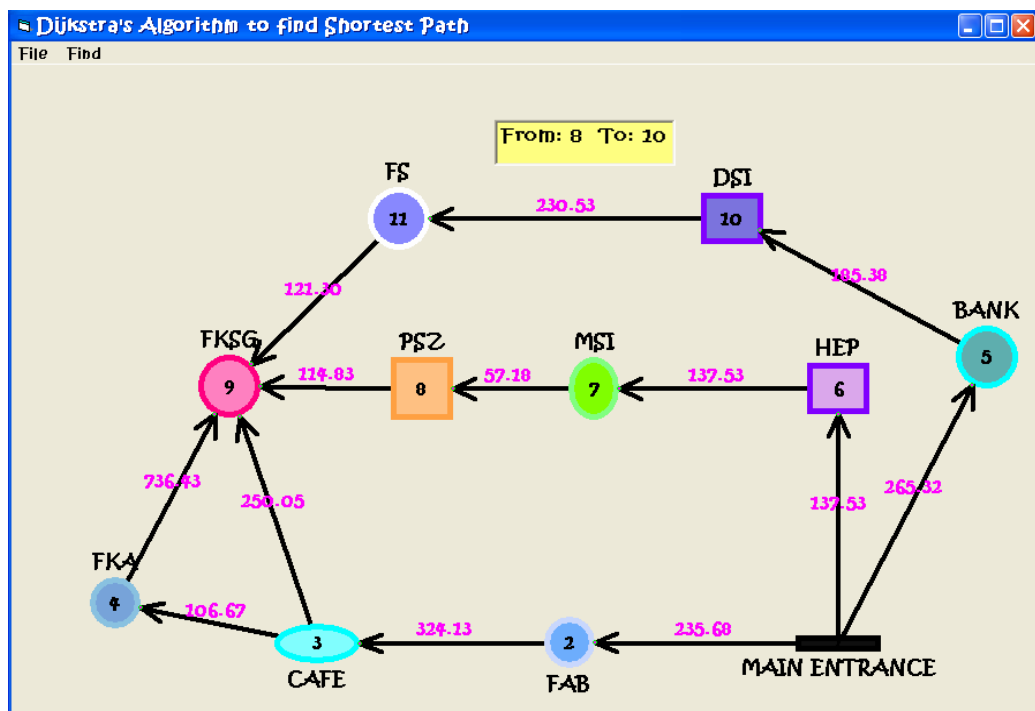
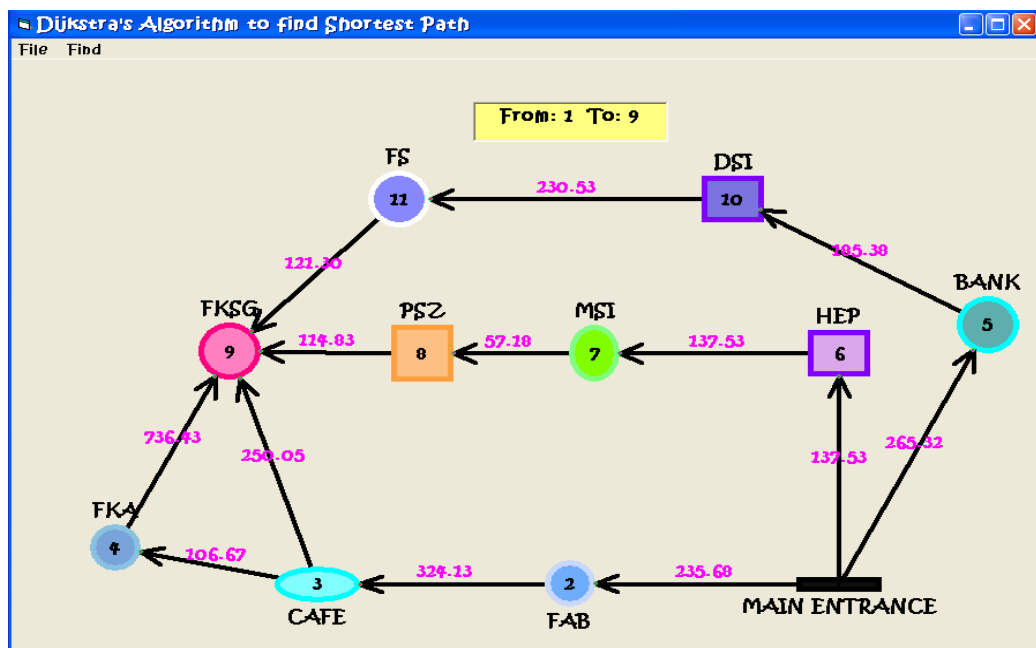
Figure 5: The methodology to implement the Dijkstra's algorithm

Part of UTM campus area as in **Figure 1** has been selected as a local network in this simulation test. **Figure 6** shows the simulation the selected route network based on **Figure 1**. Every shape represents the building that is available in the network. The numbers on each arrow indicates the distance (in meter) from a building to another. Each shape has it unique identity which was assigned in the code. The result and analysis of the simulation are discussed in the next section.

Two simulation tests were conducted to analyze the Dijkstra's algorithm:

Test 1: The origin and destination of route network was set at node 1 (Main Entrance) and at node 9 (FKSG), respectively. The configuration of the Test 1 can be seen in **Figure 6**. This test was carried out to test the ability of the Dijkstra's algorithm on positive length (i.e. multi-direction).

Test 2: The origin and destination of route network was set at node 8 (PSZ) and at node 10 (DSI), respectively. The configuration of the Test 1 can be seen in **Figure 7**. This test was carried out to test the ability of the Dijkstra's algorithm on negative length (i.e. one-direction).



In Test 1, the Dijkstra's algorithm works efficiently on finding the shortest-paths from a given origin to destination nodes. The simulation test carried out from the node 1(Main Entrance) to the node 9 (FKSG). According to **Figure 8**, there are 4 optional routes to get to the destination. **Table 1** shows the routing and the distance for each route (i.e. from the origin to the destination) that available in the network. **Figure 9** shows that the shortest-path to get to the destination is through node 6 (HEP), node 7 (MSI), and node 8 (PSZ) with a total distance of 447m.

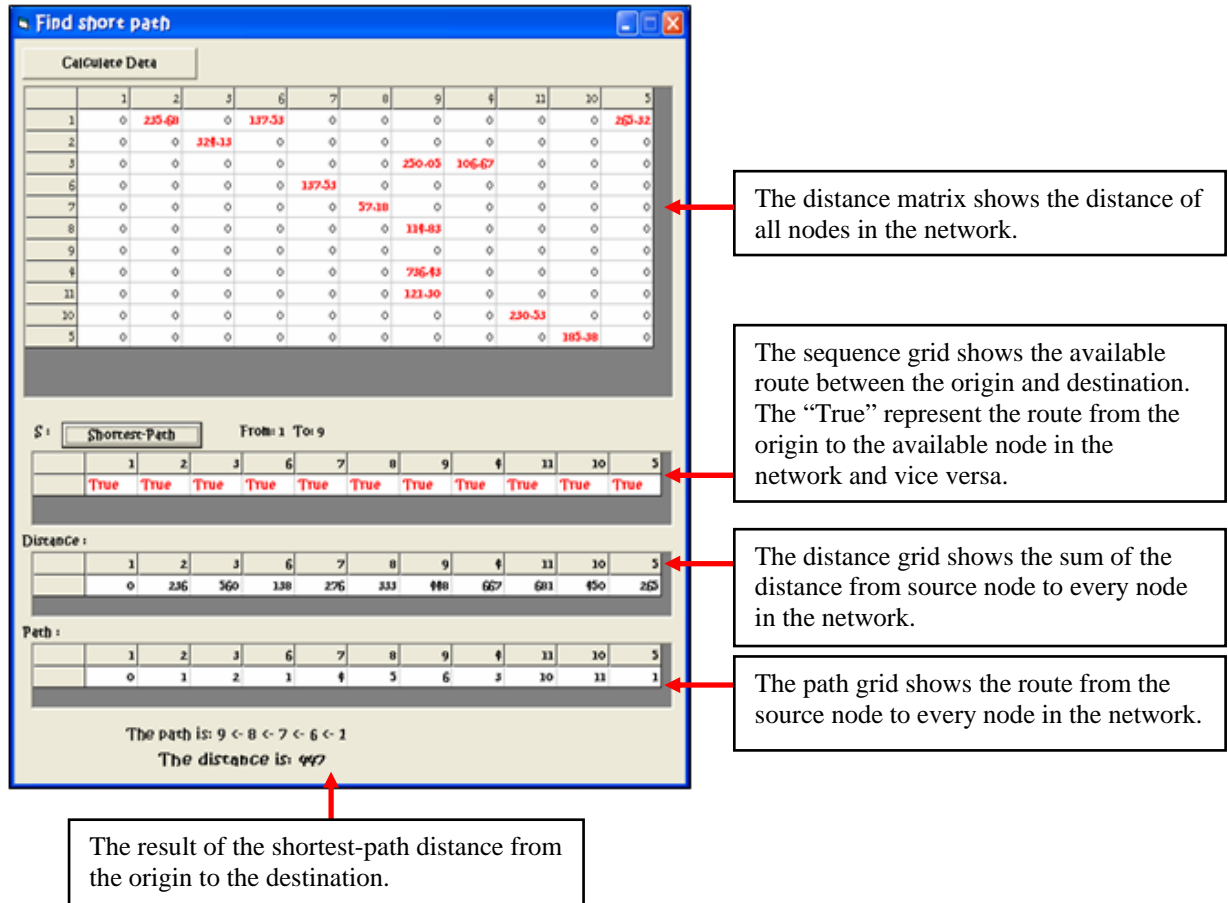


Figure 8: The result of Test 1

Route (node)	Distance (meter)
1 > 2 > 3 > 9	916.53 ~ 917
1 > 2 > 3 > 4 > 9	1402.91 ~ 1403
1 > 6 > 7 > 8 > 9	447.07 ~ 447
1 > 5 > 10 > 11 > 9	802.53 ~ 803

Table 1: The routing and the distance from the origin to the destination

In Test 2, the Dijkstra's algorithm does not work with the negative length. The result of the calculation shows that there is no shortest-path being generated since the path from node 8 to node 10 is inadmissible. **Figure 10** shows the sum of the distance from source node to every node in the network as infinity since the route is not available. Both of the results in Test 1 and Test 2 have shown the Dijkstra's algorithm and the VB codes can perform well in the network analysis.

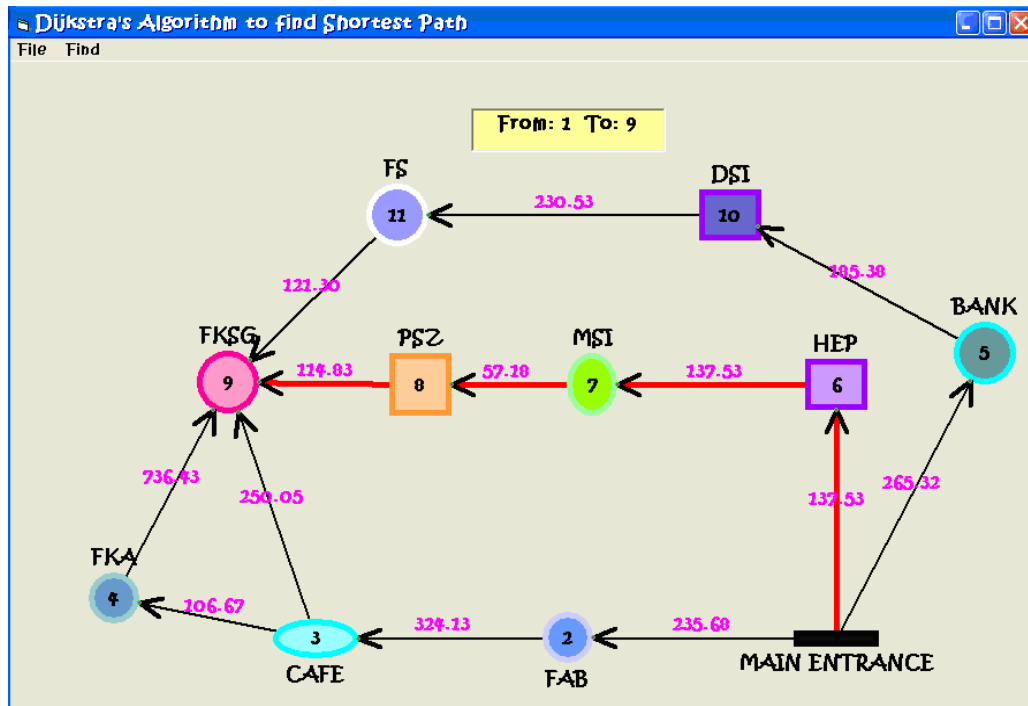
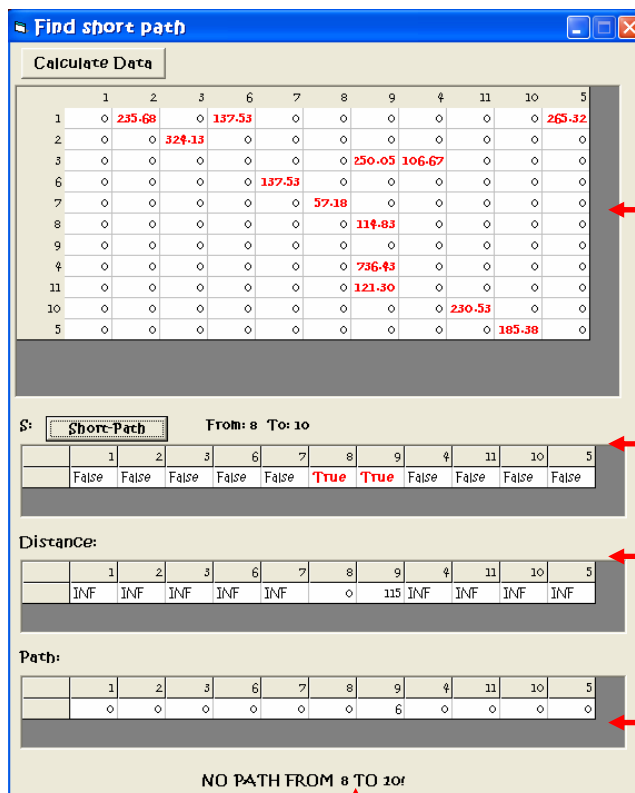


Figure 9: The shortest-path from the origin to the destination



The distance matrix shows the distance of the whole node in the network.

The sequence grid shows the available route between the origin and destination.

The distance grid shows the sum of the distance from source node to every node in the network. INF represented the distance is infinity since the route is not available.

The path grid shows the route from the source node to every node in the network.

The result shows that there is no path to from node 10 from node 8.

Figure 10: The result of Test 2

6. CONCLUSIONS & FURTHER WORKS

The Dijkstra's algorithm is capable of working with the high level of efficiency in finding the shortest-path between single source nodes to every other node in the network. However for the second simulation test, this shortest-path approaches not able to find the shortest-path in the negative length. Further works will includes the implementation of Dijkstra's algorithm in UTMnav and to extend the simulation test by applying the Floyd Warshall and A* algorithm. The simulation test will be carried out to evaluate the degree of complexity for each algorithm in computing the shortest-path. Moreover, the understanding of these simulation tests will motivate this research to improve the network analysis for UTMnav.

7. ACKNOWLEDGEMENT

This study is a part of the first author's master degree research. The first author would like to show her gratitude to her supervisors for their concerns, patience and encouragement throughout her research. She also would like to extend her appreciation to her Navigation Team for their opinion, attention and support in carrying out this study. This research is part of research vote 76012.

REFERENCES

- Ali, N. (2007). *Pembangunan Pangkalan Data dan Peta Pandu Arah Untuk Kawasan Penempatan Pelajar*. BSc Thesis. Universiti Teknologi Malaysia, Skudai.
- Alsuwaiyel, M.H. (1999). *Algorithms Design Technique and Analysis*. World Scientific Publishing, Singapore.
- Anjang, M. (2001). *Analisis algoritma laluan terpendek untuk sistem pandu kenderaan*. MSc Thesis. Universiti Teknologi Malaysia, Skudai.
- Cormen, T. H., Leiserson, C. E., & Rivest, R. (1989). *Introduction to algorithms*. McGraw-Hill, New York.
- Earnshaw, R. A. (1985). *Fundamental Algorithms for Computer Graphics*. Springer-Verlag, Berlin.
- Hofmann, W.B., Legat, K. & Weiser, M. (2003). *Principles of Positioning and Guidance*. Springer Wien New York, Austria.
- Knuth, D. E. (1973). *The Art of Computer Programming*. Addison-Wesley, Upper Saddle River.
- Nadi, S. & Delavar, M. R. (2008). *Design and development of a route guidance system for mobile devices*. MAP ASIA 2008. 18-20 August 2008. Kualau Lumpur.
- Ng, F. (2008). *Towards the Development of Universiti Teknologi Malaysia Real Time Navigation (UTMnav) System*. BSc Thesis. Universiti Teknologi Malaysia, Skudai.
- Pearl J. (1984). *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Addison-Wesley.

Russell, S. J. & Norvig, P. (2003). *Artificial Intelligence : A Modern Approach*. Prentice Hall, Upper Saddle River.

So, Y. U., Lee, K. J., Viswanathan, M., Yang, Y. K. & Whangbo, T. K. (2005). *Sketch Map Generation Method using Leveled Spatial Indexing Technique in a Mobile Environment*. IEEE.

Xu, J.J. & Chen, H. (2003). *Fighting Organized Crimes: Using Shortest-Path Algorithms to Identify Associations in Criminal Networks*. Elsevier B. V., USA.

Yeung, K.W.A. & Hall, G.B. (2007). *Spatial Database Systems*. Springer, The Netherlands.